CLAIMS

I claim:

1. In a computer system, a method for finding near identities in a DNA database, the method comprising the steps of:

providing a first database and a second database;

generating for the first database a first tag array and for the second database a second tag array; and

comparing the first tag array to the second tag array using a comparison model to determine areas of the first database which match areas of the second database.

2. The method of claim 1, wherein the first database is a genomic DNA sequence database and the second database is a genomic DNA sequence database.

3. The method of claim 1, wherein the first database is a genomic DNA sequence database and the second database is a cDNA sequence database.

4. The method of claim 1, wherein the first database is a cDNA sequence database and the second database is a cDNA sequence database.

5. The method of claim 1, wherein the step of generating for the first database a first tag array and for the second database a second tag array further comprises steps of generating a tag record which contains a tag value, a value representing a sequence ID of a sequence from which the tag value was generated and a value representing a position on a sequence from which the tag value was generated and storing the tag record in an appropriate tag array.

6. The method of claim 5, wherein the tag value is computed as

$$T = \sum_{i=1}^{|\mathrm{DNA}|} I\left(\mathrm{DNA}_i\right) \cdot 4^{(i-1)} \bmod P$$

where *T is the tag value*

DNA is a fragment of a DNA sequence,

$|DNA|$ is a length of the DNA fragment,

P is a prime number such that $P \bullet 4$ can be stored in one computer word

and where $I(DNA_i)$ evaluates to 0, 1, 2, and 3 when DNA_i is A, C, G, and T respectively.

7. The method of claim 1, wherein the step of comparing the first tag array to the second tag array using a comparison model to determine areas of the first database which match areas of the second database further comprises steps of sorting the first tag array on tag value to produce a sorted first tag array, and sorting the second tag array on tag value to produce a sorted second tag array.

8. The method of claim 7, further comprising steps of comparing each tag of each sequence of length l from the sorted first tag array to tags in the sorted second tag array and for those tag values that are equal, recording the tag values and their respective sequence ID and tag position on the sequence values in a matched tag array.

9. The method of claim 8, further comprising steps of using the matched tag array to calculate a match density value for a sequence, where the match density is equal to a total number of tags for the sequence in the matched tag array divided by a total number of tags for the sequence in the sorted second tag array, and using the match density to indicate sequences of near identity when the match density is greater than an indicator value.

10. The method of claim 9, wherein the indicator value is 0.9.

11. The method of claim 8, further comprising steps of using the matched tag array to calculate a mean difference offset value for a pair of sequences; wherein a set of offsets are differences between sequence positions associated with pairs of matching tags, and wherein a median offset is a median value of the set of offsets, and

wherein a set of differences comprises differences between the median value and each of the offsets, and wherein a mean difference offset value is a mean value of the set of differences, whereby sequences of near identity are indicated when the mean difference offset value is less than an indicator value.

12. The method of claim 11, wherein the indicator value is 25.

13. The method of claim 11, further comprising steps of using the matched tag array to calculate a rank correlation coefficient for a pair of sequences, wherein the rank correlation coefficient is computed as

$$r_s = 1 - \frac{6\left(\sum_{i=1}^{m} d_i^2\right)}{m\left(m^2 - 1\right)}$$

where r is the rank correlation coefficient for a pair of sequences,

d_i is a difference in rank of the tags, and

m is a number of tag matches, and

wherein sequences of near identity are indicated when the rank correlation coefficient for a pair of sequences is less than an indicator value.

14. The method of claim 13, wherein the indicator value is 0.75.

15. A method for creating a unique DNA genome database, comprising the steps of:

providing available genomic sequence data in a first database

enumerating regions of identity between genomic sequences in the first data base and other genomic sequences in the first database as if the first database was also a query database, the enumerating being done on a computer having a processor, memory, input/output mechanisms; and

removing from the first database, genomic sequences that are nearly identical to a region of a longer genomic sequence, whereby a unique DNA genome database is created.

16. A computer system, having a processor, memory, external data storage, input/output mechanisms, a display, for finding near identities in a DNA database, comprising:

a first database and a second database;

logic mechanisms in the computer for generating for the first database a first tag array and for the second database a second tag array; and

a comparing mechanism in the computer for comparing the first tag array to the second tag array using a comparison model to determine areas of the first database which match areas of the second database.

17. The computer system of claim 16, wherein the first database is a genomic DNA sequence database and the second database is a genomic DNA sequence database.

18. The computer system of claim 16, wherein the first database is a cDNA sequence database and the second database is a cDNA sequence database.

19. The computer system of claim 16, wherein the first database is a genomic DNA sequence database and the second database is a cDNA sequence database.

20. The computer system of claim 16, wherein the logic mechanisms for generating for the first database a first tag array and for the second database a second tag array further comprises logic mechanisms for generating a tag record which contains a tag value, a value representing a sequence ID of a sequence from which the tag value was generated and a value representing a position on a sequence from which the tag value was generated and a logic mechanism for storing the tag record in an appropriate tag array.

21. The computer system of claim 20, wherein the tag value is computed as

$$T = \sum_{i=1}^{|\mathrm{DNA}|} I(\mathrm{DNA}_i) \cdot 4^{(i-1)} \bmod P$$

where *T is the tag value*

DNA is a fragment of a DNA sequence,

$|DNA|$ is a length of the DNA fragment,

P is a prime number such that $P \bullet 4$ can be stored in one computer word

and where $I(DNA_i)$ evaluates to 0, 1, 2, and 3 when DNA_i is A, C, G, and T respectively.

22. The computer system of claim 16, wherein the comparing mechanism for comparing the first tag array to the second tag array using a comparison model to determine areas of the first database which match areas of the second database further comprises logic mechanisms for sorting the first tag array on tag value to produce a sorted first tag array, and for sorting the second tag array on tag value to produce a sorted second tag array.

23. The computer system of claim 22, further comprising logic mechanisms for comparing each tag of each sequence of length l from the sorted first tag array to tags in the sorted second tag array and for those tag values that are equal, recording the tag values and their respective sequence ID and tag position on the sequence values in a matched tag array.

24. The computer system of claim 23, further comprising a logic mechanism for using the matched tag array to calculate a match density value for a sequence, where the match density is equal to a total number of tags for the sequence in the matched tag array divided by a total number of tags for the sequence in the sorted second tag array, and using the match density to indicate sequences of near identity when the match density is greater than an indicator value.

25. The computer system of claim 24, wherein the indicator value is 0.9.

26. The computer system of claim 23, further comprising a logic mechanism for using the matched tag array to calculate a mean difference offset value for a pair of sequences; wherein a set of offsets are differences between sequence positions associated with pairs of matching tags, and wherein a median offset is a median value of the set of offsets, and wherein a set of differences comprises differences between the median value and each of the offsets, and wherein a mean difference offset value is a mean value of the set of differences, whereby sequences of near identity are indicated when the mean difference offset value is less than an indicator value.

27. The computer system of claim 26, wherein the indicator value is 25.

28. The computer system of claim 23, further comprising a logic mechanism for using the matched tag array to calculate a rank correlation coefficient for a pair of sequences, wherein the rank correlation coefficient is computed as

$$r_s = 1 - \frac{6\left(\sum_{i=1}^{m} d_i^2\right)}{m\left(m^2 - 1\right)}$$

where r is the rank correlation coefficient for a pair of sequences,

d_i is a difference in rank of the tags, and

m is a number of tag matches, and

wherein sequences of near identity are indicated when the rank correlation coefficient for a pair of sequences is less than an indicator value.

29. The computer system of claim 28, wherein the indicator value is 0.75.

30. A system for creating a unique DNA genome database, comprising;

means for providing available genomic sequence data in a first database

means for enumerating regions of identity between genomic sequences in the first data base and other genomic sequences in the first database as if the first database was also a query database, the enumerating being done on a computer having a processor, memory, input/output mechanisms; and

means for removing from the first database, genomic sequences that are nearly identical to a region of a longer genomic sequence, whereby a unique DNA genome database is created.

31. A computer program product stored on a computer readable medium for finding near identities in a DNA sequence database, comprising:

a first code mechanism for comparing a DNA sequence on a query database to DNA sequences on a data database, wherein a tag array (designated as Qtags) is generated for each of the DNA sequences on the query database and wherein a tag array (designated Dtags) is generated for each of the DNA sequences on the data database; and

a second code mechanism for comparing each Qtag to each Dtag, using a comparison model, wherein near identities of sequences in the two databases are identified.

32. The computer program product of claim 31, further comprising a code mechanism for using data which indicate that two sequences contain near identities for purposes of creating a unique DNA genome database.

33. The computer program product of claim 31, further comprising a code mechanism for using data which indicate that two sequences contain near identities for purposes of assembling two EST sequences containing near identities (a matched pair) into a single sequence (an assembled EST) which is a mathematical union of the elements of the two ESTs in the matched pair, whereby this assembled EST can replace the matched pair on the data file.

34. The computer program product of claim 31, further comprising a code mechanism for using data which indicate that two sequences contain near identities for purposes of mapping EST sequences, assembled EST sequences and cDNA sequences onto genomic sequences.

35. The computer program product of claim 34, further comprising a code mechanism for comparing sequences from the query file to sequences from the data file to determine near identities in respective data and query sequences by using a rank correlation coefficient of the sequences calculated as follows:

$$r_s = 1 - \frac{6\left(\sum_{i=1}^{m} d_i^2\right)}{m\left(m^2 - 1\right)}$$

where r is the rank correlation coefficient for a pair of sequences,

d_i is a difference in rank of the tags, and

m is a number of tag matches, and

wherein sequences of near identity are indicated when the rank correlation coefficient for a pair of sequences is less than an indicator value.

36. The computer program product of claim 35, wherein the indicator value is 0.75.

37. The computer program product of claim 31, further comprising a code mechanism for using data which indicate that two sequences contain near identities for purposes of identifying two cDNA sequences as likely alternate splices of the same gene.

38. A computer program for finding near identities in a DNA sequence database, comprising:

a first code mechanism for comparing a DNA sequence on a query database to DNA sequences on a data database, wherein a tag array (designated as Qtags) is generated for each of the DNA sequences on the query database and wherein a tag array (designated Dtags) is generated for each of the DNA sequences on the data database; and

a second code mechanism for comparing each Qtag to each Dtag, using a comparison model, wherein near identities of sequences in the two databases are identified.